

# Tutorial de enlace entre Formulario Web y Servicio Web REST

---

## Programación en Internet

Angel Manuel Gamaza Domínguez

José Miguel Otte Sainz-Aguirre



Grado en Ingeniería Informática

20 de septiembre de 2016



# Índice

<b>Índice de figuras</b>	<b>4</b>
<b>1. Creación del formulario en HTML</b>	<b>6</b>
1.1. Explicación básica sobre etiquetas y atributos . . . . .	6
1.2. Ejemplo de formulario básico en HTML . . . . .	8
<b>2. Creación del script en lenguaje JavaScript</b>	<b>12</b>
2.1. Breve introducción a la inserción de scripts en HTML . . . . .	12
2.2. Creación de la función para insertar un nuevo estudiante . . . . .	12
2.2.1. Recogida de datos desde el formulario HTML . . . . .	12
2.2.2. Función completa incorrecta en JavaScript . . . . .	13
2.2.3. Transformación a JSON de variables y Script completo correcto . . . .	14
<b>3. Opcional: crear un selector para nuestro formulario HTML</b>	<b>16</b>



## Índice de figuras

1.	Ejemplo de placeholder. . . . .	7
2.	Vista del formulario HTML. . . . .	10
3.	Ejemplo de selector en HTML . . . . .	16



# 1. Creación del formulario en HTML

## 1.1. Explicación básica sobre etiquetas y atributos

Para crear un formulario en HTML se usa la etiqueta `<form>`. Dentro de esta etiqueta se incluyen todos los campos que contendrá el formulario a rellenar.

- Un formulario se compone de varias etiquetas de tipo `<input>` acompañadas, generalmente, de otra etiqueta de tipo `<label>`.
- La etiqueta `<label>` define una etiqueta o título para la etiqueta `<input>` siguiente, de forma que en el documento HTML generado aparezca justo encima describiendo la acción a realizar.
- La etiqueta `<input>` se usa para recoger información de parte del usuario. Esta etiqueta tiene varios atributos, a través de los cuales se pueden definir unas propiedades para ella. Se explicarán los más importantes a usar de cara al proyecto realizado en la asignatura.
  - **Atributo Type:** Define el tipo de input que se va a realizar o tipo de dato que se va a recoger. Los valores más comunes para este atributo son:
    - **Text:** se recibirá texto plano.
    - **Number:** se recibirá un dato de tipo numérico.
    - **Date:** se recibirá una fecha.
    - **Password:** se recibirá una contraseña.
    - **Button:** el input será un botón para realizar una determinada acción.
  - **Atributo id:** Se usa para dotar al input de un identificador único.

Ejemplo:

```
<input id="identificador"/>
```

Esto nos creará una entrada cuyo identificador único será "identificador", a través del cual podremos referirnos a él globalmente.

- **Atributo value:** Se usa para dar un valor por defecto al input.

Ejemplo:

```
<input type="text" value="Texto plano"/>
```

Esto nos creará una entrada de tipo texto que por defecto tendrá insertada la cadena "Texto plano".

- **Atributo placeholder:** Se usa para establecer un texto o dato predeterminado en nuestra entrada, de forma que al pulsar desaparezca. Esto es muy útil para indicar qué se desea que el usuario inserte en la entrada.

Ejemplo:

```
<input type="text" placeholder="Inserte sus apellidos aquí"/>
```

Esto nos creará una entrada de tipo texto que mostrará el texto "Inserte sus apellidos aquí", lo cual desaparecerá cuando el usuario pulse y se disponga a escribir.



Figura 1: Ejemplo de placeholder.

- **Atributo onClick:** Quizás el más importante de todos. Se usa para indicar que acción realizar cuando se pulse sobre el input. Generalmente, se indica una función a la que acceder alojada en un script de tipo JavaScript.

Ejemplo:

```
<input type="button" value="Pulsa aquí" onClick="funcion()"/>
```

Esto nos creará una entrada de tipo botón que, cuando sea pulsada, nos redirigirá a un función llamada "función" donde se ejecutarán las instrucciones oportunas para dicho botón.



## 1.2. Ejemplo de formulario básico en HTML

- Dicho esto, ya podemos disponernos a crear nuestro primer formulario:

```
<form>
  <!-- Entrada correspondiente al nombre -->
  <label>Nombre</label>
  <input type="text" id="nombre" placeholder="Nombre"/>
  <br>

  <!-- Entrada correspondiente al primer apellido -->
  <label>Primer Apellido</label>
  <input type="text" id="apellido1" placeholder="Primer Apellido"/>
  <br>

  <!-- Entrada correspondiente al segundo apellido -->
  <label>Segundo Apellido</label>
  <input type="text" id="apellido2" placeholder="Segundo Apellido"/>
  <br>

  <!-- Entrada correspondiente al titulo del proyecto -->
  <label>Titulo del proyecto</label>
  <input type="text" id="titulo" placeholder="Titulo del Proyecto"/>
  <br>

  <!-- Entrada correspondiente al estado del proyecto -->
  <label>Estado del proyecto</label>
  <input type="text" id="estado" placeholder="Estado del Proyecto"/>
  <br>

  <!-- Entrada correspondiente a la fecha de presentacion -->
  <label>Fecha de presentacion</label>
  <input type="date" id="fecha"/>
  <br>

  <!-- Entrada correspondiente a la calificacion del proyecto -->
  <label>Calificacion</label>
  <input type="number" id="calificacion" placeholder="Calificacion"/>
  <br>

  <!-- Boton para realizar la accion que se desee -->
  <input type="button" value="Insertar" onClick="insertar()"/>
</form>
```

Este sería un ejemplo de formulario válido para recoger los datos necesarios para la inserción mediante un "PUT" en nuestro Servicio Web un nuevo estudiante. Al pulsar sobre el botón con el texto "Insertar" se llamará a la función "insertar()", donde se realizarán las acciones necesarias para ello.

## Creando un archivo HTML básico para insertar el formulario:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Insertar Estudiante</title>
</head>

<body>
  <form>
    <!-- Entrada correspondiente al nombre -->
    <label>Nombre</label>
    <input type="text" id="nombre" placeholder="Nombre"/>
    <br>
    <!-- Entrada correspondiente al primer apellido -->
    <label>Primer Apellido</label>
    <input type="text" id="apellido1" placeholder="Primer Apellido"/>
    <br>
    <!-- Entrada correspondiente al segundo apellido -->
    <label>Segundo Apellido</label>
    <input type="text" id="apellido2" placeholder="Segundo Apellido"/>
    <br>
    <!-- Entrada correspondiente al primer tutor -->
    <label>Tutor 1</label>
    <input type="text" id="tutor1" placeholder="Primer Tutor"/>
    <br>
    <!-- Entrada correspondiente al segundo tutor -->
    <label>Tutor 2</label>
    <input type="text" id="tutor2" placeholder="Segundo Tutor"/>
    <br>
    <!-- Entrada correspondiente al titulo del proyecto -->
    <label>Titulo del proyecto</label>
    <input type="text" id="titulo" placeholder="Titulo del Proyecto"/>
    <br>
    <!-- Entrada correspondiente al estado del proyecto -->
    <label>Estado del proyecto</label>
    <input type="text" id="estado" placeholder="Estado del Proyecto"/>
    <br>
    <!-- Entrada correspondiente a la fecha de presentacion -->
    <label>Fecha de presentacion</label>
    <input type="date" id="fecha"/>
    <br>
    <!-- Entrada correspondiente a la calificacion del proyecto -->
    <label>Calificacion</label>
    <input type="number" id="calificacion" placeholder="Calificacion"/>
    <br>
    <!-- Boton para realizar la accion que se desee -->
    <input type="button" value="Insertar" onClick="insertar()"/>
  </form>

  <!-- Cabecera de tipo h3 para informar al usuario -->
  <h3 id="responseJS"></h3>
  <script src="http://code.jquery.com/jquery-latest.js"></script>
</body>
</html>
```

Guardarlo con el nombre que se desee sin olvidar la extensión ".html" para indicar que es un archivo de este tipo.

Con esto tendremos nuestro documento HTML creado con un formulario básico, al que se puede dar formato y estilo mediante otros mecanismos pero que no son objeto de este tutorial.



The image shows a web browser window with a single tab titled "Insertar Estudiante". The address bar displays a URL that has been redacted with a thick red line, followed by "/form.html". Below the browser window, there is an HTML form with the following fields and labels:

- Nombre: [Input field with placeholder text "Nombre"]
- Primer Apellido: [Input field with placeholder text "Primer Apellido"]
- Segundo Apellido: [Input field with placeholder text "Segundo Apellido"]
- Tutor 1: [Input field with placeholder text "Primer Tutor"]
- Tutor 2: [Input field with placeholder text "Segundo Tutor"]
- Titulo del proyecto: [Input field with placeholder text "Titulo del Proyecto"]
- Estado del proyecto: [Input field with placeholder text "Estado del Proyecto"]
- Fecha de presentacion: [Input field with placeholder text "dd / mm / aaaa"]
- Calificacion: [Input field with placeholder text "Calificacion"]

At the bottom left of the form is a button labeled "Insertar".

Figura 2: Vista del formulario HTML.



## 2. Creación del script en lenguaje JavaScript

### 2.1. Breve introducción a la inserción de scripts en HTML

Existen dos formas de crear un script y añadirlo a nuestro archivo HTML.

- Mediante la etiqueta `<script>` y su definición en el propio archivo HTML.

Sintaxis:

```
<script type="text/javascript">
    /* Funciones a implementar en lenguaje JavaScript */
</script>
```

- Mediante la adición al documento de un fichero JavaScript externo con extensión ".js".

Sintaxis:

Archivo HTML: Añadir la línea siguiente:

```
<script type="text/javascript" src="archivo.js"></script>
```

Archivo JavaScript:

```
/* Definir las funciones siguiendo la sintaxis de JavaScript */
```

Es necesario que el archivo ".js" se encuentre en la misma carpeta o directorio que el archivo ".html" para usar la sintaxis anterior. Podemos crearnos una carpeta llamada ".js" donde insertar todos nuestros scripts (es lo más correcto) y escribir la ruta de dicha carpeta en el atributo *src* de la etiqueta `<script>`.

### 2.2. Creación de la función para insertar un nuevo estudiante

#### 2.2.1. Recogida de datos desde el formulario HTML

Para este caso usaremos la segunda forma de las dos indicadas anteriormente, ya que nos permite una mejor estructuración del código en general. Hemos definido en nuestro archivo HTML que cuando se pulse sobre el input que contiene el botón, se nos dirija a la función "insertar()", con lo cual implementaremos esta función en nuestro archivo ".js".

Para acceder a los diferentes campos del formulario podemos usar la función `getElementById` y escoger el campo "value".

Ejemplo:

```
var nombre = document.getElementById('nombre').value;
```

Con esto conseguimos almacenar el valor recogido por el input con id "nombre" del documento HTML en una variable de JavaScript. Lo mismo podemos hacer con todos los input de forma que tendremos fácilmente recogidos todos los valores del formulario:

```
function insertar(){
    var nombre = document.getElementById('nombre').value;
    var apellido1 = document.getElementById('apellido1').value;
    var apellido2 = document.getElementById('apellido2').value;
    var tutor1 = document.getElementById('tutor1').value;
    var tutor2 = document.getElementById('tutor2').value;
    var titulo = document.getElementById('titulo').value;
    var estado = document.getElementById('estado').value;
    var fecha = document.getElementById('fecha').value;
    var calificacion = document.getElementById('calificacion').value;
}
```

Con esto, tendríamos recogidos todos los valores de los inputs de nuestro formulario en variables JavaScript. Una vez hecho esto, solo nos queda usar la función de AJAX explicada en clase para realizar la llamada al Servicio Web, aunque con un pequeño matiz a cambiar para que funcione correctamente que se explicará después.

### 2.2.2. Función completa incorrecta en JavaScript

```
function insertar(){
    var nombre = document.getElementById('nombre').value;
    var apellido1 = document.getElementById('apellido1').value;
    var apellido2 = document.getElementById('apellido2').value;
    var tutor1 = document.getElementById('tutor1').value;
    var tutor2 = document.getElementById('tutor2').value;
    var titulo = document.getElementById('titulo').value;
    var estado = document.getElementById('estado').value;
    var fecha = document.getElementById('fecha').value;
    var calificacion = document.getElementById('calificacion').value;
    $.ajax({
        type : "POST", //El metodo para insertar seria de tipo POST.
        url : "URL del Servicio Web",
        contentType : "application/json", //Recibimos un tipo JSON.
        data : JSON.stringify({
            "nombre" : nombre,
            "apellido1" : apellido1,
            "apellido2" : apellido2,
            "tutor1" : tutor1,
            "tutor2" : tutor2,
            "titulo" : titulo,
            "estado" : estado,
            "fecha" : fecha,
            "calificacion" : calificacion
        }),
        success : function(data) {
            $("#responseJS").html("Insercion realizada.");
        },
        error : function(res) {
            alert("ERROR " + res.statusText);
        }
    });
}
```

### 2.2.3. Transformación a JSON de variables y Script completo correcto

A simple vista la anterior función en JavaScript podría parecer válida para resolver nuestro ejercicio, pero al crear el JSON a partir de los valores de las variables recogidas se producirían algunos errores. Para ello, se utiliza la función de JavaScript "toJSON", la cual puede usarse de la siguiente forma:

```
var nombre = {toJSON: function () {return document.getElementById('nombre').value;}};
```

Realizando esto, ya podríamos usar la variable para crear nuestro JSON sin ningún tipo de problema, quedando nuestro archivo JavaScript de la siguiente forma:

```
function insertar(){
    var nombre = {toJSON: function () {return document.getElementById('
        nombre').value;}};
    var apellido1 = {toJSON: function () {return document.getElementById('
        apellido1').value;}};
    var apellido2 = {toJSON: function () {return document.getElementById('
        apellido2').value;}};
    var tutor1 = {toJSON: function () {return document.getElementById('
        tutor1').value;}};
    var tutor2 = {toJSON: function () {return document.getElementById('
        tutor2').value;}};
    var titulo = {toJSON: function () {return document.getElementById('
        titulo').value;}};
    var estado = {toJSON: function () {return document.getElementById('
        estado').value;}};
    var fecha = {toJSON: function () {return document.getElementById('
        fecha').value;}};
    var calificacion = {toJSON: function () {return document.
        getElementById('calificacion').value;}};
    $.ajax({
        type : "POST", //El metodo para insertar seria de tipo POST.
        url : "URL del Servicio Web",
        contentType : "application/json",
        data : JSON.stringify({
            "nombre" : nombre,
            "apellido1" : apellido1,
            "apellido2" : apellido2,
            "tutor1" : tutor1,
            "tutor2" : tutor2,
            "titulo" : titulo,
            "estado" : estado,
            "fecha" : fecha,
            "calificacion" : calificacion
        }),
        success : function(data) {
            $("#responseJS").html("Insercion realizada.");
        },
        error : function(res) {
            alert("ERROR " + res.statusText);
        }
    });
}
```

Hecho esto, faltaría incluir la línea con el enlace al script JavaScript en nuestro archivo ".html":

```
<script type="text/javascript" src="archivo.js"></script>
```

Y ya tendríamos realizado nuestro archivo HTML con su correspondiente formulario para recoger los datos y llamar al método del Servicio Web con ellos.



### 3. Opcional: crear un selector para nuestro formulario HTML

Cuando alguno de los campos de nuestro formulario solo va a tomar una serie de valores distintos, podemos crear un selector para poder elegir alguno de ellos y transferirlos al archivo JavaScript. Esto se consigue usando la etiqueta `<select>` y sus correspondientes `<option>`.

Ejemplo:

```
<label>Estado del proyecto</label>
<select id="estado">
  <option>En desarrollo</option>
  <option>Presentado</option>
</select>
```

Con esto, conseguiríamos tener dos opciones para elegir el estado del proyecto sin tener que introducirlo manualmente.



Figura 3: Ejemplo de selector en HTML



## Referencias

- [1] FORO DE PREGUNTAS Y RESPUESTAS, Stackoverflow *<http://stackoverflow.com/>*  
Fecha de último acceso: 28 de Noviembre de 2015
- [2] THE WORLD'S LARGEST WEB DEVELOPER SITE *<http://www.w3schools.com/>*  
Fecha de último acceso: 28 de Noviembre de 2015